Solving Feature Sparseness in Text Classification using Core-Periphery Decomposition

Xia Cui, Sadamori Kojaku, Naoki Masuda, and **Danushka Bollegala**
Department of Computer Science, University of Liverpool
Department of Engineering Mathematics, University of Bristol
{xia.cui, danushka.bollegala}@liverpool.ac.uk
{sadamori.koujaku, naoki.masuda}@bristol.ac.uk

Abstract

Feature sparseness is a problem common to cross-domain and short-text classification tasks. To overcome this feature sparseness problem, we propose a novel graph decomposition-based method to find candidate features for expanding feature vectors. Specifically, we first create a feature-relatedness graph, which is subsequently decomposed into core-periphery (CP) pairs and use the peripheries as the expansion candidates of the cores. We expand both training and test instances using the computed related features and use them to train a text classifier. We observe that prioritising features that are common to both training and test instances as cores during the CP decomposition to further improve the accuracy of text classification. We evaluate the proposed CP-decomposition-based feature expansion method on benchmark datasets for cross-domain sentiment classification and short-text classification. Our experimental results show that the proposed method consistently outperforms all baselines on short-text classification tasks, and perform competitively with pivot-based cross-domain sentiment classification methods.

1 Introduction

Short-texts are abundant on the Web and appear in various different formats such as microblogs (Kwak et al., 2010), QA forums, review sites, SMS, email, and chat messages (Cong et al., 2008; Thelwall et al., 2010). Unlike lengthy responses that take time to both compose and to read, short responses have gained popularity particularly in social media contexts. Considering the steady growth of mobile devices that are physically restricted to compact keyboards, which are suboptimal for entering lengthy text inputs, it is safe to predict that the amount of short-texts will continue to grow in the future. Considering the importance and the quantity of the short-texts in various web-related tasks, such as text classification (kun Wang et al., 2012; dos Santos and Gatti, 2014), and event prediction (Sakaki et al., 2010), it is important to be able to accurately represent and classify short-texts.

Compared to performing text mining on longer texts (Guan et al., 2009; Su et al., 2011; Yogatama and Smith, 2014), for which dense and diverse feature representations can be created relatively easily, handling of shorter texts poses several challenges. The number of features that are present in a given short-text will be a small fraction of the set of all features that exist in all of the train instances. Moreover, frequency of a feature in a short-text will be small, which makes it difficult to reliably estimate the salience of a feature using term frequency-based methods. This is known as the *feature sparseness* problem in text classification.

Feature sparseness is not unique to short-text classification but also encountered in cross-domain text classification (Blitzer et al., 2006, 2007; Bollegala et al., 2014), where the training and test data are selected from different domains with small intersection of feature spaces. In the domain adaptation (DA) setting, a classifier trained on one domain (*source*) might be agnostic to the features that are unique to a different domain (*target*), which results in a *feature mismatch* problem similar to the feature-sparseness problem discussed above.

To address the feature sparseness problem encountered in short-text and cross-domain classification tasks, we propose a novel method that computes related features that can be appended to the feature vectors to reduce the sparsity. Specifically, we decompose a feature-relatedness graph into core-periphery (CP) structures, where a core feature (a vertex) is linked to a set of periph-

eries (also represented by vertices), indicating the connectivity of the graph. This graph decomposition problem is commonly known as the CP-decomposition (Csermely et al., 2013; Rombach et al., 2017; Kojaku and Masuda, 2018, 2017).

Our proposed CP-decomposition algorithm significantly extends existing CP-decomposition methods in three important ways.

- First, existing CP-decomposition methods consider unweighted graphs, whereas edges in feature-relatedness graphs are weighted (possibly nonnegative) real-valued feature-relatedness scores such as positive pointwise mutual information (PPMI). Our proposed CP-decomposition method can operate on edge-weighted graphs.

- Second, considering the fact that in text classification a particular periphery can be related to more than one core, we relax the hard assignment constraints on peripheries and allow a particular periphery attach to multiple cores.

- Third, prior work on pivot-based cross-domain sentiment classification methods have used features that are frequent in training (source) and test (target) data as expansion candidates to overcome the feature mismatch problem. Inspired by this, we define *coreness* of a feature as the pointwise mutual information between a feature and the source/target domains. The CP-decomposition algorithm we propose will then compute the set of cores considering both structural properties of the graph as well as the coreness values computed from the train/test data.

To perform feature vector expansion, we first construct a feature-relatedness graph, where vertices correspond to features and the weight of the undirected edge connecting two features represent the relatedness between those two features. Different features and relatedness measures can be flexibly used in the proposed graph construction. In our experiments, we use the simple (yet popular and effective) setting of n-gram features as vertices and compute their relatedness using PPMI. We compute the coreness of features as the sum of the two PPMI values between the feature and the source, and the feature and the target domains.[1] Next, CP-decomposition is performed on this feature-relatedness graph to obtain a set of core-periphery structures. We then rank the set of peripheries of a particular core by their PPMI values, and select the top-ranked peripheries as the expansion features of the core. We expand the core features in training and train a logistic regression-based binary classifier using the expanded feature vectors, and evaluate its performance on the expanded test feature vectors.

We evaluate the effectiveness of the proposed method using benchmark datasets for two different tasks: short-text classification and cross-domain sentiment classification. Experimental results on short-text classification show that the proposed method consistently outperforms previously proposed feature expansion-based methods for short-text classification and even some of the sentence embedding learning-based methods. Moreover, the consideration of coreness during the CP-decomposition improves the text classification accuracy. In cross-domain sentiment classification experiments, the proposed method outperforms previously proposed pivot-based methods such as the structural correspondence learning (SCL) (Blitzer et al., 2006).

2 Related Work

Two complementary approaches for overcoming feature sparseness in text classification can be identified in the literature: (a) expanding the instances by predicting the missing features, and (b) projecting the instances to a dense (potentially lower-dimensional) space and performing the classification task in this projected space. Our work can be categorised to the first group of methods. We next review prior work on both types of approaches.

Man (2014) proposed a feature vector expansion based on frequent term sets (FTS), where they first define the co-occurrence among the features and then the expansion candidates are selected by a pre-defined threshold on frequency. Finally, the features in the original feature vectors are expanded using these frequently co-occurring features. Ma et al. (2016) proposed an improvement based on FTS by introducing the support and confidence to the co-occurrence relationship when

[1] In short-text classification experiments, coreness is computed using unlabelled training and test instances.

they create the frequent term sets for expansion.

Our proposed method is related to the pivot selection methods proposed in prior work on unsupervised cross-domain sentiment classification, where common features (called the pivots) are first identified using some heuristic measure, and predictors are learnt that can accurately predict those pivots using the other (non-pivot) features. For example, in spectral feature alignment (SFA) (Pan et al., 2010), a bipartite graph is created between non-pivots (domain-specific) and pivots (domain-independent) then spectral methods are used to learn a projection from domain-specific to domain-independent feature spaces. Blitzer et al. (2006) proposed the frequency (FREQ) of a feature in the source and the target domain as the criterion for selecting pivots for structural correspondence learning (SCL) when performing cross-domain named entity recognition. However, they found (Blitzer et al., 2007) that mutual information (MI) to be a better pivot selection criterion for cross-domain sentiment classification tasks. Bollegala et al. (2015) proposed a feature expansion-based domain adaptation method, where a sentiment sensitive thesaurus (SST) is built using the pointwise mutual information (PMI) between a feature and the source/target domains. The cores identified by CP-decomposition can be seen as playing the role of pivots in cross-domain text classification tasks because cores get expanded by their corresponding peripheries during the feature expansion step. However, one notable characteristic in the proposed method is that we induce cores via CP-decomposition instead of applying heuristic measures such as MI or PMI. As we later see in the experiments, the proposed method outperforms the previous pivot-based feature expansion methods in cross-domain sentiment classification benchmarks.

A complementary approach to overcome feature-sparseness is to learn a (potentially lower dimensional) dense feature representation for the training and test instances that suffer from feature sparseness, and train and evaluate classifiers in this dense feature space instead of the original sparse feature space. Skip-thought vectors (Kiros et al., 2015) encodes a sentence into a lower-dimensional dense vector using bidirectional long short-term memory (bi-LSTM), whereas FastSent (Hill et al., 2016) learns sentence embeddings by predicting the words in the adjacent sentences in a corpus, ignoring the word ordering. Paragraph2Vec (Le and Mikolov, 2014) jointly learns sentence and word embeddings that can mutually predict each other in a short-text such as a paragraph in a document. Sequential Denoising Autoencoder (SDAE) (Hill et al., 2016) transforms an input sentence into an embedding by a look-up table consisting of pre-trained word embeddings and attempts to reconstruct the original sentence embedding from a masked version. Sentence embedding learning methods such as skip-thought vectors, FastSent, SDAE etc. require a large amount of unlabelled texts for training such as 80 million sentence Toronto books corpus, which might not be available for specialised domains. As shown in our experiments, the proposed methods perform competitively with these embedding-based methods, while not requiring any additional training data, other than the small (typically less than 50,000 sentences) benchmark training datasets.

In the CP-decomposition problem, one seeks a partition of vertices into two groups called a core and a periphery. The core vertices are densely interconnected and the peripheral vertices are sparsely interconnected. The core and peripheral vertices may be densely interconnected or sparsely interconnected. Various algorithms have been developed to find a single core-periphery structure (Csermely et al., 2013; Rombach et al., 2017) or multiple core-periphery structures (Kojaku and Masuda, 2018, 2017) in a graph. Many existing algorithms assume that each vertex belongs to only one core-periphery structure. This assumption is problematic for text classification because a peripheral vertex can belong to multiple core-periphery structures. To circumvent this problem, here we present a novel algorithm for the CP-decomposition that allows a peripheral vertex to belong to more than one core-periphery structures. Some existing CP-decomposition algorithms allow peripheral vertices to belong to multiple core-periphery structures (Yan and Luo, 2016; Sardana and Bhatnagar, 2016; Xiang et al., 2018). These algorithms detect non-overlapping communities (i.e., groups of densely interconnected vertices) in a graph. Then, they regard vertices that do not belong to any community as peripheral vertices. Therefore, the detected peripheries might not be strongly related to the associated cores because they are not densely interconnected with the

cores in general. Another CP-decomposition algorithm allows communities to overlap and regard the vertices belonging to many communities as a core (Yang and Leskovec, 2014). Then, the detected peripheral vertices may be densely interconnected because they belong to the same community. In contrast to these algorithms, the present algorithm seeks peripheries that are densely interconnected with the associated cores while sparsely interconnected with other peripheral vertices.

To the best of our knowledge, we are the first to apply CP-decomposition to any NLP task, let alone short-text classification. Moreover, our formulation of the CP-decomposition is customised to the needs in the NLP domain such as prioritising linguistically appropriate cores and allows a single periphery to link to multiple cores. We hope that our work will inspire NLP practitioners to use CP-decomposition in related NLP tasks such as information retrieval (Mihalcea and Radev, 2011) (measuring similarity between short-text documents), query suggestion/expansion (Fang, 2008) (suggesting related peripheral terms to a query corresponding to a core).

3 CP-decomposition-based Feature Expansion

Our proposed method consists of three steps: (a) building a feature-relatedness graph (Section 3.1), (b) performing CP-decomposition on the feature-relatedness graph (Sections 3.2 and 3.3) and (c) using the core-peripheries from the decomposition to perform feature expansion (Section 3.4). Next, we describe each of those steps in detail.

3.1 Feature-Relatedness Graph

Given a set of texts, we build a feature-relatedness graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathbf{W})$, where $\mathcal{V}$ is the set of vertices corresponding to the features, $\mathcal{E}$ is the set of undirected edges between two vertices in $\mathcal{G}$ and the weight of the edge $e_{ij} \in \mathcal{E}$ connecting two features i and j is given by the W_{ij} element of the weight matrix $\mathbf{W}$. Let us denote the number of vertices and edges respectively by N and M (i.e. $|\mathcal{V}| = N$ and $|\mathcal{E}| = M$). Different types of features such as n-grams, part-of-speech sequences, named entities, dependency relations etc. can be used as vertices in the feature-relatedness graph. Moreover, different relatedness measures such as co-occurrence frequency, pointwise mutual information, χ^2, log-likelihood ratio etc. can be used to compute the weights assigned to the edges. For simplicity, in this paper, we represent each text-document using the set of unigrams extracted from that document, and use PPMI to compute a non-negative $\mathbf{W}$. We connect two words if PPMI values between them are greater than zero. This formulation is used for both short-text classification and cross-domain sentiment classification experiments conducted in the paper.

3.2 Core-Periphery Decomposition

Given a feature-relatedness graph $\mathcal{G}$ created using the process described in Section 3.1, we propose a method that decomposes $\mathcal{G}$ into a set of overlapping core-periphery structures. A core-periphery structure assumed in this study consists of one core vertex and an arbitrarily number of peripheral vertices that are adjacent (i.e., directly connected) to the core vertex.[2] Therefore, a core-periphery structure forms a star graph. We further assume that a core belongs only to one core-periphery structure, but a periphery can belong to multiple core-periphery structures.

Let $\mathcal{C} \subseteq \mathcal{V}$ be the set of cores and P_i be the set of peripheries associated with the core $i(\in \mathcal{C})$. We regard that a core-periphery structure is a good pair if the core is adjacent to its peripheries with large edge weights. One goodness measure is the sum of edge weights between the core i and peripheries, which is given by $\sum_{j \in \mathcal{P}_i} W_{ij}$. This quantity should be larger than the value expected from a null model (i.e., randomised graph) for the detected core-periphery structure to be meaningful. We seek $\mathcal{C}$ and $\mathcal{P}_i$ ($\forall i \in C$) by maximising

$$Q = \sum_{i \in \mathcal{C}} \sum_{j \in \mathcal{P}_i} W_{ij} - \sum_{i \in \mathcal{C}} \sum_{j \in \mathcal{P}_i} \mathbb{E}[W_{ij}], \quad (1)$$

where $\mathbb{E}[W_{ij}]$ is the expected edge weight between vertices i and j in the null model. The first term on the right-hand side of (1) is the total weights of the edges between the cores and peripheries. The second term is the expected value of the first term according to the null model. Therefore, a large positive Q value indicates that cores and peripheries are connected with large edge weights. To compute $\mathbb{E}[W_{ij}]$, we must specify a null model. We consider a simple null model where any pair of vertices is adjacent by an edge with an equal

[2] In the remainder of the paper, we refer to core vertices as cores and peripheral vertices as peripheries to simplify the terminology.

expected weight (Erdős and Rényi, 1959). Then, we can rewrite (1) as

$$Q = \sum_{i \in \mathcal{C}} \sum_{j \in \mathcal{P}_i} (W_{ij} - p), \tag{2}$$

where p is the average edge weight of the original graph given by

$$p = \frac{2}{N(N-1)} \sum_{i,j \in \mathcal{V}, i \neq j} W_{ij}. \tag{3}$$

We maximise Q as follows. Given a set of cores $\mathcal{C}$, it is easy to find peripheries that maximise Q. Suppose a core i and a vertex $j \notin \mathcal{P}_i$, which may belong to one or more different core-periphery structures. Adding the vertex j to $\mathcal{P}_i$ increases Q, if $W_{ij} - p$ is positive. Therefore, $\mathcal{P}_i$ associated with core i must be the neighbours of vertex i with an edge weight of $W_{ij} > p$. Therefore, we have

$$\max_{\mathcal{C}} \max_{\mathcal{P}_i, i \in \mathcal{C}} Q = \max_{\mathcal{C}} \sum_{i \in \mathcal{C}} \sum_{j \in \mathcal{V} \setminus \mathcal{C}} \tilde{W}_{ij}, \tag{4}$$

where

$$\tilde{W}_{ij} = \begin{cases} W_{ij} - p & W_{ij} - p > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

(4) indicates that the maximisation of Q is equivalent to partitioning of the set of vertices $\mathcal{V}$ into $\mathcal{C}$ and $\mathcal{V} \setminus \mathcal{C}$ such that the sum of edge weights given by (5) between $\mathcal{C}$ and $\mathcal{V} \setminus \mathcal{C}$ is maximised. This is known as the max-cut problem (Goemans and Williamson, 1995). However, solving the max-cut problem is NP-hard (Karp, 1972). Therefore, we use the Kernighan-Lin's algorithm (Kernighan and Lin, 1970) to find a good (but generally a suboptimal) solution.

It should be noted that Q is conserved when we regard $\mathcal{V} \setminus \mathcal{C}$ as the cores and $\mathcal{C}$ as peripheries. This is because Q is the sum of edge weights between $\mathcal{C}$ and $\mathcal{V} \setminus \mathcal{C}$. For example, suppose a graph with a single core-periphery structure as shown in Figure 1(a). By regarding the core as a periphery and vice versa, we have another assignment of the core-periphery structure achieving the same Q value as shown in Figure 1(b). Although Q is the same in the two assignments, we would like to prioritise the core-periphery structure shown in Figure 1(a), because we would like to have a smaller set of cores than peripheries. Therefore, we regard $\mathcal{C}$ as the set of cores if $|\mathcal{C}| < |\mathcal{V} \setminus \mathcal{C}|$; otherwise we regard $\mathcal{C}$ as the set of peripheries.

3.3 Semi-supervised Core-Periphery Decomposition

The objective given by (4) depends only on $\mathcal{G}$ and does not consider any prior linguistic knowledge that we might have about which features are appropriate as cores. For example, for cross-domain sentiment classification, it has been shown that features that express similar sentiment in both source and target domains are suitable as pivots (Blitzer et al., 2007). To incorporate this information, we integrate the *coreness* of words into the objective as follows:

$$\max_{\mathcal{C}} \max_{\mathcal{P}_i, i \in \mathcal{C}} Q = \max_{\mathcal{C}} \sum_{i \in \mathcal{C}} \sum_{j \in \mathcal{V} \setminus \mathcal{C}} \tilde{W}_{ij} + \lambda \sum_{i \in \mathcal{C}} \text{coreness}(i). \tag{6}$$

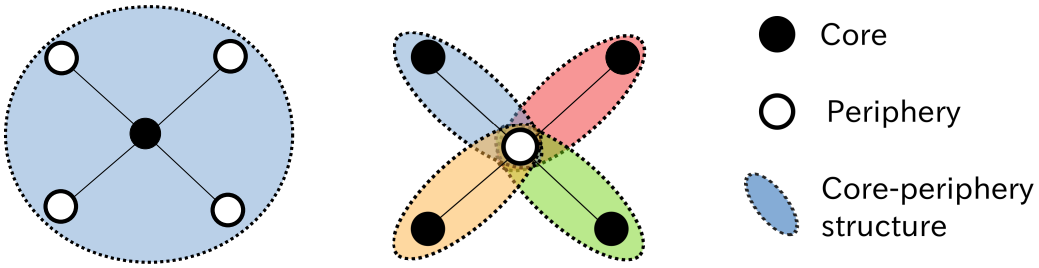
In (6), coreness(i) is a nonnegative value that indicates the appropriateness of i as a core. Hyperparameter λ adjusts the importance we would like to give to coreness as opposed to determining cores based on the graph structure. We tune λ using a held out portion of the training data in our experiments. Different measures can be used to precomputed the coreness values from the train/test data such as FREQ, MI, PMI, PPMI etc, which have been proposed in prior work on DA for selecting pivots (Blitzer et al., 2006, 2007; Bollegala et al., 2015). In this work, we use PPMI to precompute the coreness for a word i as follows:

$$\text{coreness}(i) = (\text{PPMI}(i, \mathcal{D}_{\text{train}}) - \text{PPMI}(i, \mathcal{D}_{\text{test}}))^2. \tag{7}$$

Here, $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{test}}$ are respectively the set of training and test data (or in the case of DA selected from the source and the target domains).

3.4 Feature Expansion

To overcome feature-sparseness in training and test instances, we expand features that are cores by their corresponding peripheral sets. Specifically, for each core $i \in \mathcal{C}$, we sort its peripheries $\mathcal{P}_i$ by their coreness values and select the top-k ranked peripheries as the expansion features for a core i if it appears in a document. The values of these expansion features are set to their PPMI values with the corresponding core after ℓ_1 normalising over the set of expansion features in each instance. The effect of k on performance is experimentally studied later.



(a) (b)

Figure 1: Core-periphery structures with an equal quality, Q. Each filled and empty circles indicate core and peripheral vertices, respectively. Each shared region indicates a core-periphery structure.

Dataset	SCL					CP-decomposition		
	FREQ	MI	PMI	PPMI	No Expansion	Non-overlapping	Overlapping w/o coreness	Overlapping w/ coreness
TR	67.60	66.12	67.44	63.21	78.86	80.34	80.56	**80.86**
CR	77.85	74.83	78.52	75.50	80.87	83.89	83.89	**84.40**
SUBJ	87.65	82.15	85.65	82.75	88.05	89.75	90.15	**90.48**
MR	64.68	58.07	64.26	59.10	73.55	75.23	74.95	**75.66**
AVG	74.45	70.29	73.97	70.14	80.33	82.30	82.39	**82.85**

Table 1: Results for the short-text classification task. For each dataset, the best results are shown in bold.

4 Experiments

We evaluate the proposed method on two tasks: short-text classification (a non-DA task) and cross-domain sentiment classification (a DA task). For short-text classification we use the Stanford sentiment treebank (TR)[3], customer reviews dataset (CR) (Hu and Liu, 2004), subjective dataset (SUBJ) (Pang and Lee, 2004) and movie reviews (MR) (Pang and Lee, 2005). For DA we use Amazon multi-domain sentiment dataset (Blitzer et al., 2007) containing product reviews from four categories: Books (B), DVDs (D), Electronics (E) and Kitchen Appliances (K). Each category is regarded as a domain and has 1000 positive and 1000 negative reviews, and a large number of unlabelled reviews.[4] We train a classifier on 12 domain pairs adapting from source to target (S-T): B-D, B-E, B-K, D-B, D-E, D-K, E-B, E-D, E-K, K-B, K-D, K-E. For the short-text classification datasets, we use the official train/test split.

We represent each instance (document) using a bag-of-features consisting of unigrams. Stop words are removed using a standard stop words list. We train an ℓ_2 regularised binary logistic regression classifier with each dataset, where the regularisation coefficient is tuned via 5-fold cross validation.

[3]https://nlp.stanford.edu/sentiment/treebank.html

[4]Blitzer et al. (2007) considered 4 and 5 star rated reviews as positive and 1 or 2 as negative in sentiment.

4.1 Classification Accuracy

We use the classification accuracy on the test data (i.e. ratio between the number of correctly classified test instances and the total number of test instances in the dataset) as the performance evaluation measure. As baselines we evaluate the classification accuracy without expanding features (**No Expansion**), expanding the features by a non-overlapping version of the CP-decomposition method where a single periphery will be assigned to only a single core, overlapping CP-decomposition with and without the consideration of coreness (described respectively in Sections 3.2 and 3.3). We apply SCL with pivots selected from four different criteria (FREQ, MI, PMI and PPMI) for each S-T pair in the DA datasets. Strictly speaking, SCL is a DA method but if we can apply to short-text classification tasks as well if we consider training and test datasets respectively as a source and a target domain and select pivots using some selection criterion. Results on the short-text and DA tasks are summarised respectively in Tables 1 and 2.

As shown in Table 1, all variants of the CP-decomposition outperform the **No Expansion** baseline and the best performance is reported by the overlapping CP-decomposition considering the coreness values. According to binomial test results, there is no statistical significance in Table 1. SCL performs poorly on this non-DA task, indicat-

S-T	SCL					CP-decomposition		
	FREQ	MI	PMI	PPMI	No Expansion	Non-overlapping	Overlapping w/o coreness	Overlapping w/ coreness
B-D	72.75	65.50	71.50	69.25	75.00	75.75	**76.75**	76.38
B-E	72.75	71.00	**74.50**	66.00	71.00	71.00	69.75	69.75
B-K	77.25	64.00	**80.50**	77.25	78.25	78.25	77.75	78.00
D-B	71.00	53.00	66.25	65.50	74.00	74.25	74.25	**75.25**
D-E	72.00	67.00	72.75	**74.75**	**74.75**	73.75	73.00	**74.75**
D-K	**79.75**	57.50	79.00	76.75	79.25	78.00	79.25	79.25
E-B	62.75	57.25	66.25	60.25	**69.50**	68.50	68.75	68.75
E-D	64.50	62.75	65.50	62.75	73.25	71.75	73.25	**73.50**
E-K	82.00	77.75	81.25	79.50	**84.25**	84.00	82.50	84.00
K-B	65.75	52.50	68.00	68.75	**70.00**	**70.00**	69.75	69.50
K-D	67.25	53.75	66.75	68.50	72.75	72.00	72.75	**73.63**
K-E	77.25	74.50	74.50	74.75	79.00	79.75	79.00	**80.50**
AVG	72.08	63.04	72.23	70.33	75.08	74.75	74.73	**75.27**

Table 2: Results for DA tasks. For each S-T pair, the best results are shown in bold. The last row shows the average of performance over the 12 S-T pairs.

Methods	**TR**	**CR**	**SUBJ**	**MR**
No Expansion	76.31	81.54	88.05	73.35
FTS (Man, 2014)	76.47	62.41	50.15	66.83
SCL (Blitzer et al., 2006)	67.60	78.52	87.65	64.68
SFA (Pan et al., 2010)	60.08	70.13	79.00	59.57
Proposed	**80.86**	**84.40**	**90.48**	**75.66**

Table 3: Proposed vs. feature-based methods for short-text classification.

Methods	**MR**	**CR**	**SUBJ**
Skip-thought (Kiros et al., 2015)	**76.5**	80.1	**93.6**
Paragraph2Vec (Le and Mikolov, 2014)	74.8	78.1	90.5
FastSent (Hill et al., 2016)	70.8	78.4	88.7
SDAE (Hill et al., 2016)	74.6	78.0	90.8
CNN (Kim, 2014)	76.1	79.8	89.6
Proposed	75.7	**84.4**	90.5

Table 4: Proposed vs. document-level embedding-based methods for short-text classification.

ing that it is specifically customised for DA tasks.

Table 3 compares the performance of the proposed method (i.e., overlapping version of the CP-decomposition with coreness) against FTS, a previously proposed feature expansion method and DA methods such as SCL and SFA applied to short-text classification. We see that the proposed method consistently outperforms FTS, which uses frequently occurring features as expansion candidates. This result implies that frequency of a feature alone does not enable us to find useful features for expanding sparse feature vectors. The suboptimal performance of SFA and SCL for short-text classification indicates that, despite the fact that the feature-mismatch problem in DA has some resemblance to the feature-sparseness problem in short-text classification, applying DA methods to short-text classification is not effective. On the other hand, as shown in Table 2, proposed method reports equal or the best performance for 10 out of 12 domain pairs indicating that it is effective not only for short-text classification but also for DA. However, the improvements reported in Table 2 are not statistically significant (according to Clopper-Pearson confidence intervals (Clopper and Pearson, 1934) computed at $p < 0.01$), implying that CP-decomposition is less effective on DA datasets, which contain longer (on average 5-10 sentence reviews) texts.

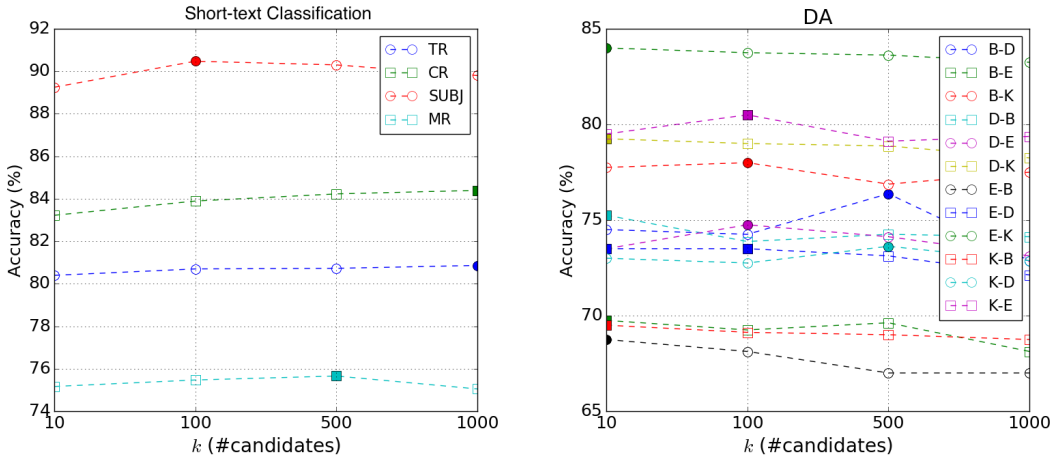
We compare the proposed method against the state-of-the-art embedding-based short-text classification methods in Table 4. For skip-thought vectors (Kiros et al., 2015), Paragraph2Vec (Le and Mikolov, 2014), FastSent (Hill et al., 2016) and SDAE (Hill et al., 2016) provided by Hill et al. (2016), we show the published results on MR, CR and SUBJ.[5] CNN represents the convolutional neural network-based document-level embedding learning method proposed by Kim (2014). The proposed method reports the best results on CR, whereas skip-thought does so for MR and SUBJ datasets. An interesting future research direction would be to combine feature-expansion method and document-level embedding methods to further improve the accuracy of short-text classification.

An example feature expansion is shown in Table 5, where 6 cores are expanded by the overlapping version of the CP-decomposition method

[5]These methods have not been evaluated on the TR dataset.

Sentence:	The film makes a strong case for the importance of the musicians in creating the motown sound.						
Methods:	Overlapping w/o coreness						Overlapping w/ coreness
Cores:	film	strong	case	musicians	creating	sound	motown
Peripheries:	tribeca	willed	neko	remixers	irritation	puget	discographer
	remakes	fliers	genitive	trombonists	populating	stereophonic	gordy
	grossing	syllabic	accusative	bandleaders	abolishing	nootka	supremes
	slasher	roderick	dative	saxophonists	duopoly	mcmurdo	stax
	blaxploitation	oxidizing	eeml	clarinetists	soundscapes	blaster	dozier

Table 5: An example of cores and top 5 peripheries chosen by overlapping CP-decomposition with/without coreness ($k = 5$). This example sentence in TR is classified incorrectly using the method without coreness (and the **No Expansion** baseline) but correctly after considering coreness.



(a) (b)

Figure 2: Number of expansion candidates for the proposed method. The marker for the best result for each dataset is filled.

without using coreness and one core with the proposed method. Top 5-ranked peripheries are shown for each core, which are used as the expansion features. We see that many cores are found without constraining the CP-decomposition by coreness, introducing noisy expansions resulting in an incorrect prediction. On the other hand, although by integrating coreness into the CP-decomposition process we have only a single matching core, *motown*, it is adequate for making the correct prediction. *motown* is a music company, which is expanded by a more general periphery *discographer*, which is a type of music performer, helping the final classification. Consideration of coreness improves the classification accuracy in both short-text classification as well as DA.

In both Tables 1 and 2, the non-overlapping version performs poorly compared to the overlapping counterpart. With non-overlapping CP-decomposition, peripheries are not allowed to connect to multiple cores. This results in producing a large number of cores each with a small number of peripheries, which does not help to overcome the feature-sparseness because each core will be expanded by a different periphery.

Figure 2 shows the effect of the number of expansion candidates k on the performance of the proposed overlapping CP-decomposition with coreness. For short-text classification (Figure 2a), the accuracy increases for $k \geq 100$ (TR and CR obtain the best for $k = 1000$). For DA (Figure 2b), $k \leq 100$ yields better performance in most of the domain pairs (10 out of 12). For all 12 domain pairs, the accuracy achieved a peak when $k \leq 500$.

5 Conclusion

We proposed a novel algorithm for decomposing a feature-relatedness graph into core-periphery structures considering coreness of a feature. Our experimental results show that the induced core-periphery structures are useful for reducing the feature-sparseness in short-text classification and cross-domain sentiment classification tasks, as indicated by their improved performance. We hope this research will encourage the society to imply

different CP decomposition methods with different tasks in NLP.

References

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proc. of ACL*. pages 440–447.

John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In *Proc. of EMNLP*. pages 120–128.

Danushka Bollegala, Takanori Maehara, and Ken ichi Kawarabayashi. 2015. Unsupervised cross-domain word representation learning. In *Proc. of ACL*. pages 730 – 740.

Danushka Bollegala, David Weir, and John Carroll. 2014. Learning to predict distributions of words across domains. In *Proc. of ACL*. pages 613 – 623.

C. J. Clopper and E. S. Pearson. 1934. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika* 26(4):404. https://doi.org/10.1093/biomet/26.4.404.

Gao Cong, Long Wang, Chin-Yew Lin, Young-In Song, and Yueheng Sun. 2008. Finding question-answer pairs from online forums. In *Proc. of SIGIR*. pages 467–474. https://doi.org/10.1145/1390334.1390415.

Peter Csermely, András London, Ling-Yun Wu, and Brian Uzzi. 2013. Structure and dynamics of core/periphery networks. *Journal of Complex Networks* 1(2):93–123.

Cicero dos Santos and Maira Gatti. 2014. Deep convolutional neural networks for sentiment analysis of short texts. In *Proc. of COLING*. pages 69–78. http://www.aclweb.org/anthology/C14-1008.

P Erdős and A Rényi. 1959. On random graphs i. *Publ. Math.* 6:290–297.

Hui Fang. 2008. A re-examination of query expansion using lexical resources. In *Proc. of ACL*. pages 139–147.

Michel X Goemans and David P Williamson. 1995. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM (JACM)* 42(6):1115–1145.

Hu Guan, Jinguy Zhou, and Minyi Guo. 2009. A class-feature-centroid classifier for text categorization. In *Proc. of WWW*. pages 201 – 210.

Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. Learning distributed representations of sentences from unlabelled data. In *Proceedings of NAACL-HLT*. pages 1367–1377.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *KDD 2004*. pages 168–177.

Richard. M. Karp. 1972. *Reducibility among Combinatorial Problems*. Springer US, Boston, MA.

Brian W Kernighan and Shen Lin. 1970. An efficient heuristic procedure for partitioning graphs. *The Bell System Technical Journal* 49(2):291–307.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. pages 1746–1751.

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in neural information processing systems*. pages 3294–3302.

Sadamori Kojaku and Naoki Masuda. 2017. Finding multiple core-periphery pairs in networks. *Physical Review E* 96(5):052313.

Sadamori Kojaku and Naoki Masuda. 2018. Core-periphery structure requires something else in the network. *New Journal of Physics* 40:043012.

Bing kun Wang, Yong feng Huang, Wan xia Yang, and Xing Li. 2012. Short text classification based on strong feature thesaurus. *Journal of Zhejiang University-SCIENCE C (Computers and Electronics)* 13(9):649 – 659.

Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. 2010. What is twitter, a social network or a news media? In *Proc. of WWW*. pages 591–600. https://doi.org/10.1145/1772690.1772751.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International Conference on Machine Learning*. pages 1188–1196.

Huifang Ma, Lei Di, Xiantao Zeng, Li Yan, and Yuyi Ma. 2016. Short text feature extension based on improved frequent term sets. In Zhongzhi Shi, Sunil Vadera, and Gang Li, editors, *Intelligent Information Processing VIII*. Springer International Publishing, Cham, pages 169–178.

Yuan Man. 2014. Feature extension for short text categorization using frequent term sets. *Procedia Computer Science* 31:663 – 670. 2nd International Conference on Information Technology and Quantitative Management, ITQM 2014.

Rada Mihalcea and Dragomir Radev. 2011. *Graph-based Natural Language Processing and Information Retrieval*. Cambridge University Press.

Sinno Jialin Pan, Xiaochuan Ni, Jian-Tao Sun, Qiang Yang, and Zheng Chen. 2010. Cross-domain sentiment classification via spectral feature alignment. In *Proc. of WWW*. pages 751–760.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, page 271.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd annual meeting on association for computational linguistics*. Association for Computational Linguistics, pages 115–124.

Puck Rombach, Mason A. Porter, James H. Fowler, and Peter J. Mucha. 2017. Core-periphery structure in networks (revisited). *SIAM Review* 59(3):619–646.

Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. 2010. Earthquake shakes twitter uers: Real-time event detection by social sensors. In *Proc. of WWW*. pages 851–860.

Divya Sardana and Raj Bhatnagar. 2016. Core periphery structures in weighted graphs using greedy growth. In *Proc. 2016 IEEE/WIC/ACM Int. Conf. Web Intelligence Core*. ACM, New York, pages 1–8.

Jiang Su, Jelber Sayyad-Shirabad, and Stan Matwin. 2011. Large scale text classification using semi-supervised multinomial naive bayes. In *Proc. of ICML*.

Mike Thelwall, Kevan Buckley, Georgios Paltoglou, Di Cai, and Arvid Kappas. 2010. Sentiment strength detection in short informal text. *Journal of the Association for Information Science and Technology* 61(12):2544–2558.

Bing-Bing Xiang, Zhong-Kui Bao, Chuang Ma, Xingyi Zhang, Han-Shuang Chen, and Hai-Feng Zhang. 2018. A unified method of detecting core-periphery structure and community structure in networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science* 28(1):013122.

Bowen Yan and Jianxi Luo. 2016. Multicore-periphery structure in networks. Preprint arXiv:1605.03286.

Jaewon Yang and Jure Leskovec. 2014. Overlapping communities explain core–periphery organization of networks. *Proc. IEEE* 102(12):1892–1902.

Dani Yogatama and Noah A. Smith. 2014. Making the most of bag of words: Sentence regularization with alternating direction method of multipliers. In *Proc. of ICML*. pages 656 – 664.